



TITLE:

Macintosh 版 Risa/Asir(数式処理における理論と応用の研究)

AUTHOR(S):

野呂, 正行

CITATION:

野呂, 正行. Macintosh 版 Risa/Asir(数式処理における理論と応用の研究). 数理解析研究所講究録 1993, 848: 196-198

ISSUE DATE:

1993-09

URL:

<http://hdl.handle.net/2433/83649>

RIGHT:

Macintosh 版 Risa/Asir

富士通国際研 野呂 正行 (Masayuki NORO)¹⁾

Abstract. UNIX 上で開発 / 動作している数式処理システム Risa/Asir を Macintosh 上に移植した。ここでは、移植作業、機能、動作環境、UNIX 版との相違点などについて述べる。

1. はじめに

Risa/Asir は、UNIX ワークステーション (WS) 上で開発が続けられているが、近年のパーソナルコンピュータ (PC) の高速化、大容量化に伴い WS、PC の差は OS のみとなり、PC 上でも十分実用に耐える数式処理システムが実現できると考えられる。代表的な PC として、MS-DOS マシンと Macintosh (Mac) があげられるが、開発環境およびメモリ空間の制限、ウィンドウシステムの使いやすさなどを考慮して、Mac への移植をまず行なった。目標としては、UNIX 版とほぼ同等の機能を持たせることである。

2. 移植の実際

Asir の移植で本質的な部分は次の 3 つである。

- 倍精度整数乗除算
- メモリ管理
- dynamic object loader

Mac 上に移植する場合、倍精度乗除算に関しては、CPU が 680X0 であるため、sun3、NEWS などのコードがほぼそのまま使用可能である。

やや問題となったのはメモリ管理である。UNIX 版では、実行中に動的にヒープを伸ばすが、Mac 版では、やや特殊な Mac のメモリマネージャとの兼ね合いもあり、ヒープは初期化時に獲得したものを固定して使用することとした。また、静的データ領域が、UNIX の場合と異なりスタックの上位の部分にあるために、marking の仕方を多少替える必要があったが、大きな変更は必要としなかった。

第三の dynamic object loader に関しては、独立した link editor が無いため現在は未着手である。

以上のように、WS に移植する場合に問題となる点はある程度容易に解決したが、実際に最も問題となった点は、ソースプログラム全体を ANSI 対応にする必要があるという

¹⁾nororo@iiias.flab.fujitsu.co.jp

点であった。使用したコンパイラは THINK-C V5.0 で、prototype 宣言をおろそかにできないため、結局 UNIX 版の Risa/Asir の全てのソースプログラムの関数に対して prototype を宣言した。これに伴い、これまで UNIX 版の C コンパイラではフリーパスであった、キャストなしの関数引数をことごとく正しい型にキャストする必要があった。

これらの作業により、とりあえず、Mac 版 Risa/Asir は動作した。この時点で UNIX 版と比較して欠けている点は、dynamic loader を除いては、

- ユーザプログラムを読み込む際の cpp フィルタ
- graphics 機能
- PARI library とのリンク

であった。いずれも必須ではないが、UNIX 版とほぼ同等の機能という目標を実現するためには不可欠である。

まず cpp であるが、これは、ある free software を asir に組み込んでしまうことで解決した。Mac 上では、子プロセスへのパイプを用いることができないので、cpp を通したプログラムを一旦 disk に書き込んで、それを読み込むことにした。

graphics 機能は、UNIX 版自体まだ実験的なもので、実現されている機能は 2 次元描画のごく一部であるが、UNIX 版で実現されている機能はほぼ実現した。ここで最も問題となったのは、ウィンドウシステムに対するイベント処理と asir 内での計算の実行の両立である。UNIX 版では、描画は別プロセスが行なっているので、asir 自体がイベントを処理する必要はないが、Mac 版では asir が計算を実行している時にもある程度、ウィンドウに関する操作が行なえる必要がある。しかし、Mac では、UNIX における signal のような割り込み機構が使えないので、計算中にも何らかのタイミングでイベントをチェックする必要がある。現在の Mac 版 asir では、ある程度頻繁に呼ばれるメモリ管理部の一部でこのチェックを行なっている。ここでは、キーボードからの割り込み (UNIX 版では Ctrl-C、Mac 版では Command-'.') もチェックしている。これにより、大きな計算に入っても、割り込みおよびウィンドウ操作が遅延することは事実上ないと言える。

第三点の PARI とのリンクは、big float および種々の特殊関数の評価を必要とする計算にとって不可欠である。残念ながら、提供されている Mac 版 PARI のソースプログラムが MPW 用のもので、THINK-C で library を作るためには数千行のアセンブラファイルを THINK-C 用に変換するという作業を行なう必要がある。そのため、これは未着手である。²⁾

3. UNIX 版との相違点 (まとめ)

UNIX 版との相違点は、短所長所まとめて現在の所次の通りである。

²⁾ 京都での発表後、PARI の portable 版のごく一部をアセンブラで書き直したものを Mac 版にリンクした。スピードは、MPW 版と比較して 1/3 ほどであるが、とりあえず、Mac 版においても PARI の機能を用いることはできるようになった。

- dynamic object loader がない。
- PARI がリンクされていない。
- ヒープの大きさ、ガベジコレクションの頻度などのパラメタは、params というファイルに記述する。
- GNU の history ライブラリを組み込んであり、UNIX の fep 風のヒストリ置き換えおよびコマンドライン編集ができる。
- mini finder によるファイルの選択ができる。

最後の、mini finder によるファイルの選択は、Mac 標準的なインタフェースでありかつ容易に実現できるため、組み込んだ。その他、ヘルプ機能など、組み込むべき機能はいくつかある。それらは今後順次組み込んでいく予定である。

4. 動作環境

Mac 版 asir のファイルサイズは約 300KB である。メモリは、1MB 程度でも動作することが確認されているが、最低 2MB 以上あることが望ましい。移植に用いたマシンは、MacIIci/System7 であるが、System6 でも動作する。また、System7 では、32bit on/off、仮想記憶 on/off いずれのモードでも動作する。SE、Classic など CPU が 68000 のマシン上では、倍精度整数除算の部分を 68000 に置き換えたものが動作する。

5. おわりに

今回の移植作業で最も時間を要したのは、ANSI 対応に書き換える作業であった。しかし、この作業により、変換したソースは UNIX 上でも ANSI コンパイラに対応できるようになった。また、UNIX 版と THINK-C 版のマージ作業もほぼ完了し、今後は一つのソースで両方に対応できるようになる予定である。さらに、もう一つの PC である MS-DOS マシンについてであるが、FM-TOWNS 用に、DOS-Extender 用の GNU C が提供されているため、ANSI 対応版のソースを使って FM-TOWNS に移植してみた。このコンパイラでは、本来 UNIX のシステムコールである brk(), sbrk() などが実現されていたため、メモリ管理部の数個のパラメタと、倍長整数乗除算の呼びだし関数を数個書くだけで、容易に動作した。graphics 機能などがどの程度提供されているか未だ不明のため、これ以上の移植作業は行っていないが、純粋 MS-DOS マシン用の petit-asir と共に、asir がある程度完全な形で 80X86 PC 上で動作する時もそう遠くないと考えている。³⁾

最後に、富士通国際研の松岡雅裕氏には Mac 上でのプログラミングに関して数々の有益な助言を頂き、また多くの資料を提供して頂いた。ここに感謝の意を表したい。

³⁾ 研究集会の後、graphics と dynamic object loader を除いて UNIX 版と同等の機能をもつ asir を DOS-Extender go32 上で動作させることに成功した。